

Reinforcement Learning Approaches to Coordination in Cooperative Multi-agent Systems

Spiros Kapetanakis¹, Daniel Kudenko¹, and Malcolm J.A. Strens²

¹ Department of Computer Science, University of York
Heslington, York, YO10 5DD, UK

{`spiros,kudenko`}@cs.york.ac.uk,

² Guidance and Imaging Solutions, QinetiQ
Ively Road, Farnborough, Hampshire GU14 0LX, UK
mjstrens@qinetiq.com

Abstract. We report on an investigation of reinforcement learning techniques for the learning of coordination in cooperative multi-agent systems. Specifically, we focus on two novel approaches: one is based on a new action selection strategy for Q-learning [10], and the other is based on model estimation with a shared action-selection protocol. The new techniques are applicable to scenarios where mutual observation of actions is not possible.

To date, reinforcement learning approaches for such *independent* agents did not guarantee convergence to the optimal joint action in scenarios with high miscoordination costs. We improve on previous results [2] by demonstrating empirically that our extension causes the agents to converge almost always to the optimal joint action even in these difficult cases.

1 Introduction

Learning to coordinate in cooperative multi-agent systems is a central and widely studied problem (e.g., [5,1,2,6,7,11]). In this context, coordination is defined as *the ability of two or more agents to jointly reach a consensus over which actions to perform in an environment*. We investigate the case of *independent* agents that cannot observe one another's actions, which often is a more realistic assumption.

In this investigation, we focus on reinforcement learning, where the agents must learn to coordinate their actions through environmental feedback. To date, reinforcement learning (RL) methods for independent agents [9,7] did not guarantee convergence to the *optimal* joint action in scenarios where miscoordination is associated with high penalties. Even approaches using agents that are able to build predictive models of each other (so-called *joint-action learners*) have failed to show convergence to the optimal joint action in such difficult cases [2].

We investigate two approaches to reinforcement learning in search of improved convergence to the optimal joint action in the case of independent agents.

The first approach is a variant of Q-learning [10] where we introduce a novel estimated value function in the Boltzmann action selection strategy. The second technique is based on a shared action-selection protocol that enables the agents to estimate the rewards for specific joint actions.

We evaluate both RL approaches experimentally on two especially difficult coordination problems that were first introduced by Claus and Boutilier in [2]: the *climbing game* and the *penalty game*. The empirical results show that the convergence probability to the optimal joint action is greatly improved over other approaches, in fact reaching almost 100%.

Our paper is structured as follows: we first introduce the aforementioned common testbed for the study of learning coordination in cooperative multi-agent systems. We then introduce each reinforcement learning technique and discuss the experimental results. We finish with an outlook on future work.

2 Single-Stage Coordination Games

A common testbed for studying the problem of multi-agent coordination is that of repeated cooperative single-stage games [3]. In these games, the agents have common interests, i.e. they are rewarded based on their joint action and all agents receive the same reward. In each round of the game, every agent chooses an action. These actions are executed simultaneously and the reward that corresponds to the joint action is broadcast to all agents.

A more formal account of this type of problem was given in [2]. In brief, we assume a group of n agents $\alpha_1, \alpha_2, \dots, \alpha_n$, each of which have a finite set of *individual actions* A_i , known as the agent's *action space*. In this game, each agent α_i chooses an individual action from its action space to perform. The action choices make up a *joint action*. Upon execution of their actions all agents receive the reward that corresponds to the joint action. For example, Table 1 describes the reward function for a simple cooperative single-stage game. If agent 1 executes action b and agent 2 executes action a , the reward they receive is 5. Obviously, the optimal joint-action in this simple game is (b, b) as it is associated with the highest reward of 10.

Table 1. A simple cooperative game.

		Agent 1	
		a	b
Agent 2	a	3	5
	b	0	10

Our goal is to enable the agents to learn optimal coordination from repeated trials. To achieve this goal, one can use either *independent* or *joint-action* learners. The difference between the two types lies in the amount of information they can perceive in the game. Although both types of learners can perceive the reward that is associated with each joint action, the former are unaware of the existence of other agents whereas the latter can also perceive the actions of others. In this way, joint-action learners can maintain a model of the strategy of other agents and choose their actions based on the other participants'

perceived strategy. In contrast, independent learners must estimate the value of their individual actions based solely on the rewards that they receive for their actions. In this paper, we focus on individual learners, these being more universally applicable.

In our study, we focus on two particularly difficult coordination problems, the climbing game and the penalty game. These games were introduced in [2]. This focus is without loss of generality since the climbing game is representative of problems with high miscoordination penalty and a single optimal joint action whereas the penalty game is representative of problems with high miscoordination penalty and multiple optimal joint actions. Both games are played between two agents. The reward functions for the two games are included in Tables 2 and 3:

Table 2. The climbing game table.

		Agent 1		
		a	b	c
Agent 2	a	11	-30	0
	b	-30	7	6
	c	0	0	5

In the climbing game, it is difficult for the agents to converge to the optimal joint action (a, a) because of the negative reward in the case of miscoordination. For example, if agent 1 plays a and agent 2 plays b , then both will receive a negative reward of -30. Incorporating this reward into the learning process can be so detrimental that both agents tend to avoid playing the same action again. In contrast, when choosing action c , miscoordination is not punished so severely. Therefore, in most cases, both agents are easily tempted by action c . The reason is as follows: if agent 1 plays c , then agent 2 can play either b or c to get a positive reward (6 and 5 respectively). Even if agent 2 plays a , the result is not catastrophic since the reward is 0. Similarly, if agent 2 plays c , whatever agent 1 plays, the resulting reward will be at least 0. From this analysis, we can see that the climbing game is a challenging problem for the study of learning coordination. It includes heavy miscoordination penalties and “safe” actions that are likely to tempt the agents away from the optimal joint action.

Another way to make coordination more elusive is by including multiple optimal joint actions. This is precisely what happens in the penalty game of Table 3.

In the penalty game, it is not only important to avoid the miscoordination penalties associated with actions (c, a) and (a, c) . It is equally important to agree on which optimal joint action to choose out of (a, a) and (c, c) . If agent 1 plays a expecting agent 2 to also play a so they can receive the maximum reward of 10 but agent 2 plays c (perhaps expecting agent 1 to play c so that, again, they receive the maximum reward of 10) then the resulting penalty can be very detrimental to both agents’ learning process. In this game, b is the “safe” action for both agents since playing b is guaranteed to result in a reward of 0 or 2, regardless of what the other agent plays. Similarly with the climbing

Table 3. The penalty game table.

		Agent 1		
		a	b	c
Agent 2	a	10	0	k
	b	0	2	0
	c	k	0	10

game, it is clear that the penalty game is a challenging testbed for the study of learning coordination in multi-agent systems.

3 A Q-Learning Approach to Learning of Coordination

A popular technique for learning coordination in cooperative single-stage games is one-step Q-learning, a reinforcement learning technique. In this section, we first introduce the general approach, followed by a discussion of the novel FMQ heuristic for action selection. We end the section with empirical results and a discussion of limitations of the FMQ approach.

3.1 Basics

Since the agents in a single-stage game are stateless, we need a simple reformulation of the general Q-learning algorithm such as the one used in [2]. Each agent maintains a Q value for each of its actions. The value $Q(\text{action})$ provides an estimate of the usefulness of performing this action in the next iteration of the game and these values are updated after each step of the game according to the reward received for the action. We apply Q-learning with the following update function:

$$Q(\text{action}) \leftarrow Q(\text{action}) + \lambda(r - Q(\text{action}))$$

where λ is the learning rate ($0 < \lambda < 1$) and r is the reward that corresponds to choosing this action.

In a single-agent learning scenario, Q-learning is guaranteed to converge to the optimal action independent of the action selection strategy. In other words, given the assumption of a stationary reward function, single-agent Q-learning will converge to the optimal policy for the problem. However, in a multi-agent setting, the action selection strategy becomes crucial for convergence to *any* joint action. A major challenge in defining a suitable strategy for the selection of actions is to strike a balance between exploring the usefulness of moves that have been attempted only a few times and exploiting those in which the agent's confidence in getting a high reward is relatively strong. This is known as the *exploration/exploitation problem*.

The action selection strategy that we have chosen for our research is the Boltzmann strategy [4] which states that agent α_i chooses an action to perform in the next iteration

of the game with a probability that is based on its current estimate of the usefulness of that action, denoted by $EV(action)^1$:

$$P(action) = \frac{e^{\frac{EV(action)}{T}}}{\sum_{action' \in A_i} e^{\frac{EV(action')}{T}}}$$

In the case of Q-learning, the agent's estimate of the usefulness of an action may be given by the Q values themselves, an approach that has been usually taken to date.

We have concentrated on a proper choice for the two parameters of the Boltzmann function: the estimated value and the temperature. The importance of the temperature lies in that it provides an element of controlled randomness in the action selection: high values in temperature encourage exploration since variations in Q values become less important. In contrast, low temperature values encourage exploitation. The value of the temperature is typically decreased over time from an initial value as exploitation takes over from exploration until it reaches some designated lower limit. The three important settings for the temperature are the initial value, the rate of decrease and the number of steps until it reaches its lowest limit. The lower limit of the temperature needs to be set to a value that is close enough to 0 to allow the learners to converge by stopping their exploration. Variations in these three parameters can provide significant difference in the performance of the learners. For example, starting with a very high value for the temperature forces the agents to make random moves until the temperature reaches a low enough value to play a part in the learning. This may be beneficial if the agents are gathering statistical information about the environment or the other agents. However, this may also dramatically slow down the learning process.

It has been shown [8] that convergence to a joint action can be ensured if the temperature function adheres to certain properties. However, we have found that there is more that can be done to ensure not just convergence to *some* joint action but convergence to the *optimal* joint action, even in the case of independent learners. This is not just in terms of the temperature function but, more importantly, in terms of the action selection strategy. More specifically, it turns out that a proper choice for the estimated value function in the Boltzmann strategy can significantly increase the likelihood of convergence to the optimal joint action.

3.2 FMQ Heuristic

In difficult coordination problems, such as the climbing game and the penalty game, the way to achieve convergence to the optimal joint action is by influencing the learners towards their individual components of the optimal joint action(s). To this effect, there exist two strategies: altering the Q-update function and altering the action selection strategy.

Lauer and Riedmiller [5] describe an algorithm for multi-agent reinforcement learning which is based on the *optimistic* assumption. In the context of reinforcement learning, this assumption implies that an agent chooses any action it finds suitable expecting the

¹ In [4], the estimated value is introduced as *expected reward* (ER).

other agent to choose the best match accordingly. More specifically, the optimistic assumption affects the way Q values are updated. Under this assumption, the update rule for playing action α defines that $Q(\alpha)$ is only updated if the new value is greater than the current one.

Incorporating the optimistic assumption into Q -learning solves both the climbing game and penalty game every time. This fact is not surprising since the penalties for miscoordination, which make learning optimal actions difficult, are neglected as their incorporation into the learning tends to lower the Q values of the corresponding actions. Such lowering of Q values is not allowed under the optimistic assumption so that all the Q values eventually converge to the maximum reward corresponding to that action for each agent. However, the optimistic assumption fails to converge to the optimal joint action in cases where the maximum reward is misleading, e.g., in stochastic games (see experiments below). We therefore consider an alternative: the *Frequency Maximum Q Value* (FMQ) heuristic.

Unlike the optimistic assumption, that applies to the Q update function, the FMQ heuristic applies to the action selection strategy, specifically the choice of $EV(\alpha)$, i.e. the function that computes the estimated value of action α . As mentioned before, the standard approach is to set $EV(\alpha) = Q(\alpha)$. Instead, we propose the following modification:

$$EV(\alpha) = Q(\alpha) + c * \text{freq}(\text{maxR}(\alpha)) * \text{maxR}(\alpha)$$

where:

- ① $\text{maxR}(\alpha)$ denotes the maximum reward encountered *so far* for choosing action α .
- ② $\text{freq}(\text{maxR}(\alpha))$ is the fraction of times that $\text{maxR}(\alpha)$ has been received as a reward for action α over the times that action α has been executed.
- ③ c is a weight that controls the importance of the FMQ heuristic in the action selection.

Informally, the FMQ heuristic carries the information of how frequently an action produces its maximum corresponding reward. Note that, for an agent to receive the maximum reward corresponding to one of its actions, the other agent must be playing the game accordingly. For example, in the climbing game, if agent 1 plays action a which is agent 1's component of the optimal joint-action (a, a) but agent 2 doesn't, then they both receive a reward that is less than the maximum. If agent 2 plays c then the two agents receive 0 and, provided they have already encountered the maximum rewards for their actions, both agents' FMQ estimates for their actions are lowered. This is due to the fact that the frequency of occurrence of maximum reward is lowered. Note that setting the FMQ weight c to zero reduces the estimated value function to: $EV(\alpha) = Q(\alpha)$.

In the case of independent learners, there is nothing other than action choices and rewards that an agent can use to learn coordination. By ensuring that enough exploration is permitted in the beginning of the experiment, the agents have a good chance of visiting the optimal joint action so that the FMQ heuristic can influence them towards their appropriate individual action components. In a sense, the FMQ heuristic defines a

model of the environment that the agent operates in, the other agent being part of that environment.

3.3 Experimental Results

This section contains our experimental results. We compare the performance of Q-learning using the FMQ heuristic against the baseline experiments i.e. experiments where the Q values are used as the estimated value of an action in the Boltzmann action selection strategy. In both cases, we use only independent learners. The comparison is done by keeping all other parameters of the experiment the same, i.e. using the same temperature function and experiment length. The evaluation of the two approaches is performed on both the climbing game and the penalty game.

Temperature Settings. Exponential decay in the value of the temperature is a popular choice in reinforcement learning. This way, the agents perform all their learning until the temperature reaches some lower limit. The experiment then finishes and results are collected. The temperature limit is normally set to zero which may cause complications when calculating the action selection probabilities with the Boltzmann function. To avoid such problems, we have set the temperature limit to 1 in our experiments².

In our analysis, we use the following temperature function:

$$T(x) = e^{-sx} * \text{max_temp} + 1$$

where x is the number of iterations of the game so far, s is the parameter that controls the rate of exponential decay and max_temp is the value of the temperature at the beginning of the experiment. For a given length of the experiment (max_moves) and initial temperature (max_temp) the appropriate rate of decay (s) is automatically derived. Varying the parameters of the temperature function allows a detailed specification of the temperature. For a given max_moves , we experimented with a variety of s , max_temp combinations and found that they didn't have a significant impact on the learning in the baseline experiments. Their impact is more significant when using the FMQ heuristic. This is because setting max_temp at a very high value means that the agent makes random moves in the initial part of the experiment. It then starts making more informed moves (i.e. moves based on the estimated value of its actions) when the temperature has become low enough to allow variations in the estimated value of an action to have an impact on the probability of selecting that action.

Evaluation on the Climbing Game. The climbing game has one optimal joint action (a, a) and two heavily penalised actions (a, b) and (b, a) . We use the settings $\text{max_temp} = 500$ and vary max_moves from 500 to 2000. The learning rate λ is set to 0.9. Figure 1 depicts the likelihood of convergence to the optimal joint action in the baseline experiments and using the FMQ heuristic with $c = 1$, $c = 5$ and $c = 10$. The FMQ heuristic outperforms the baseline experiments for all settings of c . For $c = 10$, the FMQ heuristic converges to the optimal joint action almost always even for short experiments.

² This is done without loss of generality.

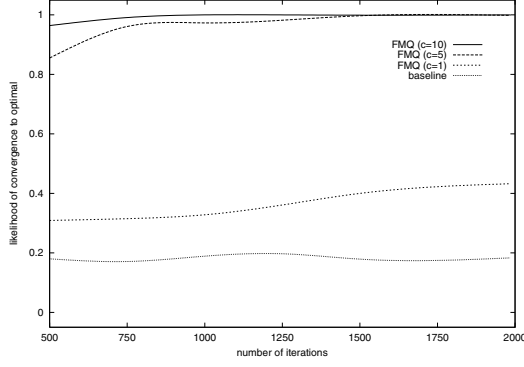


Fig. 1. Likelihood of convergence to the optimal joint action in the climbing game (averaged over 1000 trials).

Evaluation on the Penalty Game. The penalty game is harder to analyse than the climbing game. This is because it has two optimal joint actions (a, a) and (c, c) for all values of $k \leq 0$. The extent to which the optimal joint actions are reached by the agents is affected severely by the size of the penalty. However, the performance of the agents depends not only on the size of the penalty k but also on whether the agents manage to agree on which optimal joint action to choose. Figure 2 depicts the performance of the learners for $k = 0$ for the baseline experiments and with $c = 1$ for the FMQ heuristic.

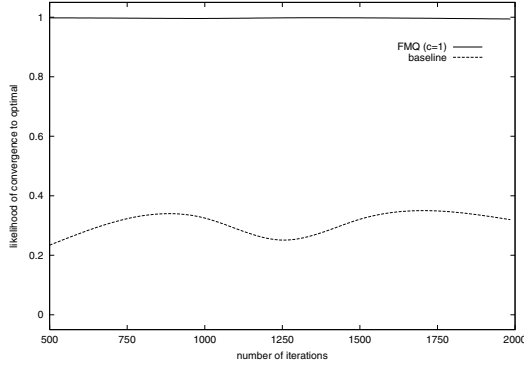


Fig. 2. Likelihood of convergence to the optimal joint action in the penalty game $k = 0$ (averaged over 1000 trials).

As shown in Figure 2, the performance of the FMQ heuristic is much better than the baseline experiment. When $k = 0$, the reason for the baseline experiment's failure is not the existence of a miscoordination penalty. Instead, it is the existence of multiple optimal joint actions that causes the agents to converge to the optimal joint action so

infrequently. Of course, the penalty game becomes much harder for greater penalty. To analyse the impact of the penalty on the convergence to optimal, Figure 3 depicts the likelihood that convergence to optimal occurs as a function of the penalty. The four plots correspond to the baseline experiments and using Q-learning with the FMQ heuristic for $c = 1$, $c = 5$ and $c = 10$.

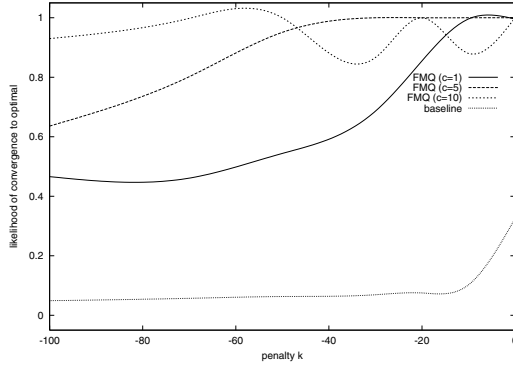


Fig. 3. Likelihood of convergence to the optimal joint action as a function of the penalty (averaged over 1000 trials).

From Figure 3, it is obvious that higher values of the FMQ weight c perform better for higher penalty. This is because there is a greater need to influence the learners towards the optimal joint action when the penalty is more severe.

3.4 Further Experiments

We have described two approaches that perform very well on the climbing game and the penalty game: FMQ and the optimistic assumption. However, the two approaches are different and this difference can be highlighted by looking at alternative versions of the climbing game. In order to compare the FMQ heuristic to the optimistic assumption [5], we introduce a variant of the climbing game which we term *the partially stochastic climbing game*. This version of the climbing game differs from the original in that one of the joint actions is now associated with a stochastic reward. The reward function for the partially stochastic climbing game is included in Table 4.

Joint action (b, b) yields a reward of 14 or 0 with probability 50%. The partially stochastic climbing game is functionally equivalent to the original version. This is because, if the two agents consistently choose their b action, they receive the same overall value of 7 over time as in the original game.

Using the optimistic assumption on the partially stochastic climbing game consistently converges to the suboptimal joint action (b, b) . This is because the frequency of occurrence of a high reward is not taken into consideration at all. In contrast, the FMQ heuristic shows much more promise in convergence to the optimal joint action. It also

Table 4. The partially stochastic climbing game table.

		Agent 1		
		a	b	c
Agent 2	a	11	-30	0
	b	-30	14/0	6
	c	0	0	5

compares favourably with the baseline experimental results. Tables 5, 6 and 7 contain the results obtained with the baseline experiments, the optimistic assumption and the FMQ heuristic for 1000 experiments respectively. In all cases, the parameters are: $s = 0.006$, $max_moves = 1000$, $max_temp = 500$ and, in the case of FMQ, $c = 10$.

Table 5. Baseline experimental results.

	a	b	c
a	212	0	3
b	0	12	289
c	0	0	381

Table 6. Results with optimistic assumption.

	a	b	c
a	0	0	0
b	0	1000	0
c	0	0	0

Table 7. Results with the FMQ heuristic.

	a	b	c
a	988	0	0
b	0	4	0
c	0	7	1

The final topic for evaluation of the FMQ heuristic is to analyse the influence of the weight (c) on the learning. Informally, the more difficult the problem, the greater the need for a high FMQ weight. However, setting the FMQ weight at too high a value can be detrimental to the learning. Figure 4 contains a plot of the likelihood of convergence to optimal in the climbing game as a function of the FMQ weight.

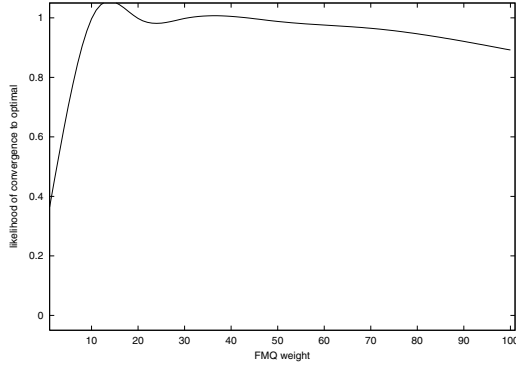


Fig. 4. Likelihood of convergence to optimal in the climbing game as a function of the FMQ weight (averaged over 1000 trials).

From Figure 4, we can see that setting the value of the FMQ weight above 15 lowers the probability that the agents will converge to the optimal joint action. This is because, by setting the FMQ weight too high, the probabilities for action selection are influenced too much towards the action with the highest FMQ value which may not be the optimal joint action early in the experiment. In other words, the agents become too narrow-minded and follow the heuristic blindly since the FMQ part of the estimated value function overwhelms the Q values. This property is also reflected in the experimental results on the penalty game (see Figure 3) where setting the FMQ weight to 10 performs very well in difficult experiments with $-100 < k < -50$ but there is a drop in performance for easier experiments. In contrast, for $c = 1$ the likelihood of convergence to the optimal joint action in easier experiments is significantly higher than in more difficult ones.

3.5 Limitations of the FMQ Approach

The FMQ heuristic performs equally well in the partially stochastic climbing game and the original deterministic climbing game. In contrast, the optimistic assumption only succeeds in solving the deterministic climbing game. However, we have found a variant of the climbing game in which both heuristics perform poorly: the *fully stochastic climbing game*. This game has the characteristic that *all* joint actions are probabilistically linked with two rewards. The average of the two rewards for each action is the same as the original reward from the deterministic version of the climbing game so the two games are functionally equivalent. For the rest of this discussion, we assume a 50% probability. The reward function for the stochastic climbing game is included in Table 8.

It is obvious why the optimistic assumption fails to solve the fully stochastic climbing game. It is for the same reason that it fails with the partially stochastic climbing game. The maximum reward is associated with joint action (b, b) which is a suboptimal action. The FMQ heuristic, although it performs marginally better than normal Q-learning still doesn't provide any substantial success ratios.

Table 8. The stochastic climbing game table (50%).

		Agent 1		
		a	b	c
Agent 2	a	10/12	5/-65	8/-8
	b	5/-65	14/0	12/0
	c	5/-5	5/-5	10/0

In the following section, we present a different reinforcement learning technique that solves the fully stochastic climbing game.

4 A Protocol-Based Reinforcement Learning Approach

In games with stochastic payoffs it is difficult to distinguish between the two sources of variation in observed payoff for some action. It would be useful to have a protocol that allows 2 or more agents to select the same joint action repeatedly in order to build up a model for the stochastic payoff distribution. This section describes a new approach for achieving this.

The basic idea is that agents follow a shared action selection policy that enables them to estimate the payoffs for each joint action. The action selection policy is based on the following idea: if an agent chooses an action at time i , then the agent is required to choose the same action at specific future time points, defined by a *Commitment Sequence*. Note, that this approach does not require agents to observe each others actions. The only assumption that the commitment sequence approach makes is that all agents share the same global clock and that they follow a common protocol for defining sequences of time-slots.

4.1 Commitment Sequences

A *commitment sequence* is some list of “time slots” (t_1, t_2, \dots) for which an agent is committed to taking the same action. If two or more agents have the same protocol for defining these sequences, then the ensemble of agents is committed to selecting a single joint-action for every time in the sequence. Although each agent does not know the action choices of the other agents, it can be certain that the observed payoffs will be statistically stationary and represent unbiased samples for the payoff distribution of *some* joint action. In order to allow a potentially infinite number of sequences to be considered as the agent learns, it is necessary that the sequences are finite *or* have an exponentially increasing time interval $\delta_i \equiv t_{i+1} - t_i$ between successive time slots. A sufficient condition is $\gamma\delta_{i+1} \geq \delta_i$ where $\gamma > 1$ for all $i > i_0$ (for some pre-defined constant i_0). In the results given here, sequences are infinite with $\gamma = 8/7$.

The successive increments are generated as follows: $\delta_{i+1} = \lfloor (8\delta_i + 6)/7 \rfloor$ where $\lfloor \cdot \rfloor$ indicates rounding down to an integer value. To ensure that no two sequences select

the same time slot, a simple mechanism is introduced. Denote the next time slot for sequence j by t_j . At time t , if all t_j are greater than t an exploratory action is chosen. Otherwise the first match (the smallest j for which $t_j = t$) is selected to determine the exploitative action. For sequence j , the increment defined above is used to update t_j . However, t_k is additionally incremented by one for all sequences *except* $k = j$. (As an alternative, it is possible to only increment by one the t_k for which $k > j$. This is a better way to keep the ratio of successive increments close to γ .)

For example, using the above function, the first commitment sequence starts with (1, 3, 6, 10, 15, 21, 28, ...). The second sequence therefore starts at time slot 2 with (2, 5, 9, 14, 20, 27, ...).

4.2 Finding the Exploitative Action

For time i suppose the agents chose actions $(a_1^i, a_2^i, \dots, a_m^i)$ (where m is the number of agents). Then an estimate of the value of this joint action is available as the average payoff received during the part of the sequence that has been completed so far. Longer sequences provide more reliable estimates. To reason about the true expected payoff, we must make some assumptions about the possible form of the stochastic payoff for each joint action: for example it must have finite variance. Here we use a Gaussian model and estimate its mean and variance from the observations. If n payoffs are observed with empirical average m and sum of squares S , we obtain estimates for the population mean μ and its variance σ_μ :

$$\hat{\mu} = m$$

$$\hat{\sigma}_\mu^2 = \frac{S + \sigma_0^2}{n^2} - \frac{m^2}{n}$$

σ_0 is a parameter to the algorithm and should be based on the expected variance of payoffs in the game; in all our experiments $\sigma_0 = 10$. In order to prefer longer sequences (more reliable estimates), a pessimistic estimate $\hat{\mu} - N_\sigma \hat{\sigma}$ is used to provide a lower bound on the expected return for each sequence. At any given time, the exploitative behaviour for an agent is to choose the action corresponding to the sequence with the greatest lower bound. Large values of N_σ reduce the risk that an optimistic bias in the payoff estimate from a short sequence will affect the choice of action. However, smaller values may give faster initial learning. In the results below, $N_\sigma = 4$.

4.3 Exploration Policy

Each agent must choose an action at the start of each sequence. A new sequence starts whenever no existing sequence is active in the current time slot. There are two obvious ways to select the new action: either explore (select the action randomly and uniformly) or exploit (select the action currently considered optimal). The simple approach used here is to choose randomly between exploration and exploitation for each sequence. For a 2-agent system, we choose the exploration probability to be $1/\sqrt{2}$. This ensures that *both* agents select an exploratory action with probability $1/2$. As an exception, the first N_{init} sequences (where $N_{init} \geq 1$) must be exploratory to ensure that an exploitative action can be calculated. In the results below, $N_{init} = 10$.

Table 9. Results for partially stochastic climbing game

	a	b	c
a	995	0	0
b	0	5	0
c	0	0	0

Table 10. Results for stochastic climbing game

	a	b	c
a	992	0	0
b	0	4	4
c	0	0	0

4.4 Experimental Evaluation

The commitment sequence method was successful for all the problems described in the previous section, including the stochastic climbing game. We tested the method over 1000 trials, with the number of moves per trial being restricted to either 500 or 1000.

In the climbing game, the likelihood of convergence to the optimal exploitative action reached 0.985 after 500 moves, i.e. the exploitative action after 500 moves was optimal in 985 of the 1000 trials. This increased to an optimal 1.000 when the number of moves was increased to 1,000. For the stochastic climbing game, the convergence probability to the optimal joint action was 0.992 after 1000 moves.

In the penalty game with 1000 moves, the commitment sequence approach always converged to an optimal joint action for all values of k between -100 and 0 . For the partially stochastic climbing game, the convergence probability to the optimal joint action was 0.995. For the stochastic climbing game, convergence probability was 0.992 after 1000 trials.

5 Outlook

We have presented an investigation of two techniques that allows two independent agents that are unable to sense each other's actions to learn coordination in cooperative single-stage games, even in difficult cases with high miscoordination penalties. However, there is still much to be done towards understanding exactly how the action selection strategy can influence the learning of optimal joint actions in this type of repeated games. In the future, we plan to investigate this issue in more detail.

Furthermore, since agents typically have a state component associated with them, we plan to investigate how to incorporate such coordination learning mechanisms in multi-stage games. We intend to further analyse the applicability of various reinforcement learning techniques to agents with a substantially greater action space. Finally, we intend

to perform a similar systematic examination of the applicability of such techniques to partially observable environments where the rewards are perceived stochastically.

References

1. C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 478–485, 1999.
2. Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.
3. Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.
4. Leslie Pack Kaelbling, Michael Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 1996.
5. Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the Seventeenth International Conference in Machine Learning*, 2000.
6. Sandip Sen and Mahendra Sekaran. Individual learning of coordination knowledge. *JETAI*, 10(3):333–356, 1998.
7. Sandip Sen, Mahendra Sekaran, and John Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426–431, Seattle, WA, 1994.
8. S. Singh, T. Jaakkola, M. L. Littman, and C Szepesvari. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning Journal*, 38(3):287–308, 2000.
9. Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337, 1993.
10. C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England, 1989.
11. Gerhard Weiss. Learning to coordinate actions in multi-agent systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 311–316. Morgan Kaufmann Publ., 1993.